Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Lightweight Rail Surface Defect Detection via Boundary-Distance Refinement and Shape-Aware Regression Reweighting

Chuang Yang
Dalian Jiaotong University
Hua Li
lihua@djtu.edu.cn

Dalian Jiaotong University

Article

Keywords: rail surface defect detection, lightweight object detection, boundary refinement, direct l/t/r/b regression, shape-aware reweighting, YOLO26

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11346423/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Lightweight Rail Surface Defect Detection via Boundary-Distance Refinement and Shape-Aware Regression Reweighting

YANG Chuang[1], LI Hua[2]*

YANG Chuang: 2011142771@qq.com; *Corresponding author: LI Hua, lihua@djtu.edu.cn

[1] Zhan Tianyou College (CRRC College), Dalian Jiaotong University, Dalian 116028, China; [2] Zhan Tianyou College (CRRC College), Dalian Jiaotong University, Dalian 116028, China

Abstract: Rail surface defects often exhibit weak contrast, elongated geometry, and ambiguous boundaries, making accurate localization difficult under class imbalance and textured rail backgrounds. This paper presents LBX-RSABR-YOLO26, a lightweight detector that couples inference-side boundary refinement with training-side regression reweighting. The LBX head aligns multiscale features, extracts differential detail cues, and introduces a zero-initialized residual branch that refines the four direct l/t/r/b distance outputs. RSABR complements this design by reweighting positive-sample regression using box area, aspect ratio, and class priors together with clipping, mean normalization, and task-assignment quality weighting. On a six-class dataset containing 7,400 images, the proposed model achieves 70.84% mAP50-95, while its training record peaks at 70.90%. Relative to YOLO26n, it reduces the parameter count from 2.6433 M to 2.5017 M and FLOPs from 6.3966 G to 6.2914 G. Ablation results identify LBX as the stronger standalone contributor, while the combined design delivers the best lightweight performance. These results demonstrate that boundary-distance refinement and shape-aware regression reweighting complement each other effectively for rail surface defect detection.

Keywords: rail surface defect detection; lightweight object detection; boundary refinement; direct l/t/r/b regression; shape-aware reweighting; YOLO26

1 Introduction

Rails directly support train loads, making their surface condition critical to operational safety. Wheel-rail contact, heavy-haul impact, temperature variation, braking heat, and environmental corrosion can produce cracks, spalling, corrugation, squats, scuffing, and wheel burns. Continued defect growth may ultimately cause rail fracture. Manual inspection is constrained by operator experience and limited maintenance windows, whereas ultrasonic and eddy-current inspection can be sensitive to equipment and environmental conditions. Recent real-time detectors, including YOLOv9, YOLOv10, RT-DETR, and YOLOv12, have advanced gradient utilization, end-to-end training, attention-based modeling, and NMS-free detection [1-3, 28]. YOLO26 further targets efficient end-to-end deployment with a simplified regression head and unified real-time vision design [16]. These developments provide a strong basis for lightweight rail-surface inspection, where both localization accuracy and computational efficiency are important.

Early visual inspection systems relied on hand-designed image-processing pipelines. Typical systems cropped a rail-surface subimage, normalized nonuniform illumination, and located defects from projected contours; later hierarchical designs fused longitudinal context with transverse priors to suppress irregular textures. These approaches were efficient and tolerant of uneven brightness and reflection, but their dependence on rail-background models and projected contours limited their ability to represent complex shapes and broken boundaries. They also left box regression for small, elongated defects largely unoptimized.

Deep-learning rail inspection has increasingly shifted toward boundary-aware, multiscale, and dense-prediction designs. Wu et al. [4] proposed a hybrid rail-surface segmentation framework that uses boundary guidance to preserve rail edges. Wang et al. [5] developed a deep-learning detector for railway surface defects with explicit attention to edge defects and multiscale variation. Wang et al. [6] introduced a dual-fusion feature pyramid and CIoU-based proposal refinement in an improved Mask R-CNN. Zhang et al. [7, 8] combined HSV-space image enhancement with an improved YOLOX detector, BiFPN feature fusion, and NAM attention, reporting a 2.42% mAP improvement at the cost of additional computation. Guo et al. [12] further demonstrated pixelwise semantic segmentation for automated rail-surface defect inspection, emphasizing the importance of fine boundary representation.

Recent studies have placed greater emphasis on lightweight deployment and high-precision localization. Wang et al. [9] improved YOLOv8n with SPD-Conv, EMA attention, and Focal-SIoU for rail-surface defects. Cao et al. [10] introduced scale-adaptive feature processing, BiFPN, a small-object head, and Wise-IoU for heavy-haul railway defects. Du et al. [11] proposed RSDNet with multiscale dilated convolution, BiFPN, and EMA attention. Transformer-based and feature-fusion approaches have also been explored: Guo et al. [19] developed RailFormer for rail-surface defect segmentation, while Zhong and Chen [20] proposed dual-path feature fusion. Zhang et al. [21] used a MobileNetV3-based YOLOv7 variant with EIoU, and Si et al. [22] proposed a Swin-Transformer-based segmentation network for multiscale defects. More recent lightweight YOLO studies include GD-YOLOv8 [23], an improved YOLO11 rail-defect detector [24], and a 2026 contact-rail framework that uses area-adaptive EIoU to improve small and elongated

defect localization [25]. Fast R-CNN-based evaluation [26] and machine-vision studies [27] likewise confirm the continued shift from handcrafted inspection toward data-driven rail-surface analysis. Taken together, the literature improves illumination robustness, background suppression, multiscale representation, segmentation quality, and deployment efficiency, but high-precision localization of weak, elongated boundaries remains challenging. This motivates the present focus on explicit correction of direct box-distance outputs and geometry-aware regression reweighting.

Motivated by these issues, we propose LBX-RSABR-YOLO26. Rather than enlarging the backbone, the method couples two lightweight mechanisms that act at different stages of detection. LBX introduces shared detail enhancement and a zero-initialized boundary residual branch that refines the four direct *l/t/r/b* distance outputs. RSABR provides complementary training supervision by adjusting positive-sample regression weights according to box area, aspect ratio, and class priors, with clipping, mean normalization, and task-assignment quality weighting used to control the overall loss scale. Thus, LBX modifies the regression representation used for prediction, whereas RSABR modifies the contribution of positive samples during optimization.

2 Proposed LBX-RSABR-YOLO26 Method

2.1 Overall Architecture

YOLO performs bounding-box regression, class prediction, and confidence estimation in a single forward pass [1, 2]. We adopt YOLO26n as the lightweight baseline, retain its backbone and PAN-FPN multiscale feature-fusion structure, and modify the detection head and training loss to address the small size, elongated shape, and indistinct boundaries of rail surface defects. As shown in Fig. 1, the resulting network comprises a backbone feature extractor, a PAN-FPN neck, the LBX detection head, and the RSABR loss.

The lightweight configuration used on the experimental platform is YOLO26n [16], while YOLO26s is used only as a larger-scale reference. Unless otherwise stated, LBX-RSABR-YOLO26 denotes the YOLO26n-scale implementation of the proposed method. Both configurations retain the multiscale feature extraction, feature fusion, and decoupled detection structure of the YOLO26 family and were retrained on the rail-defect dataset using the experimental settings described below.

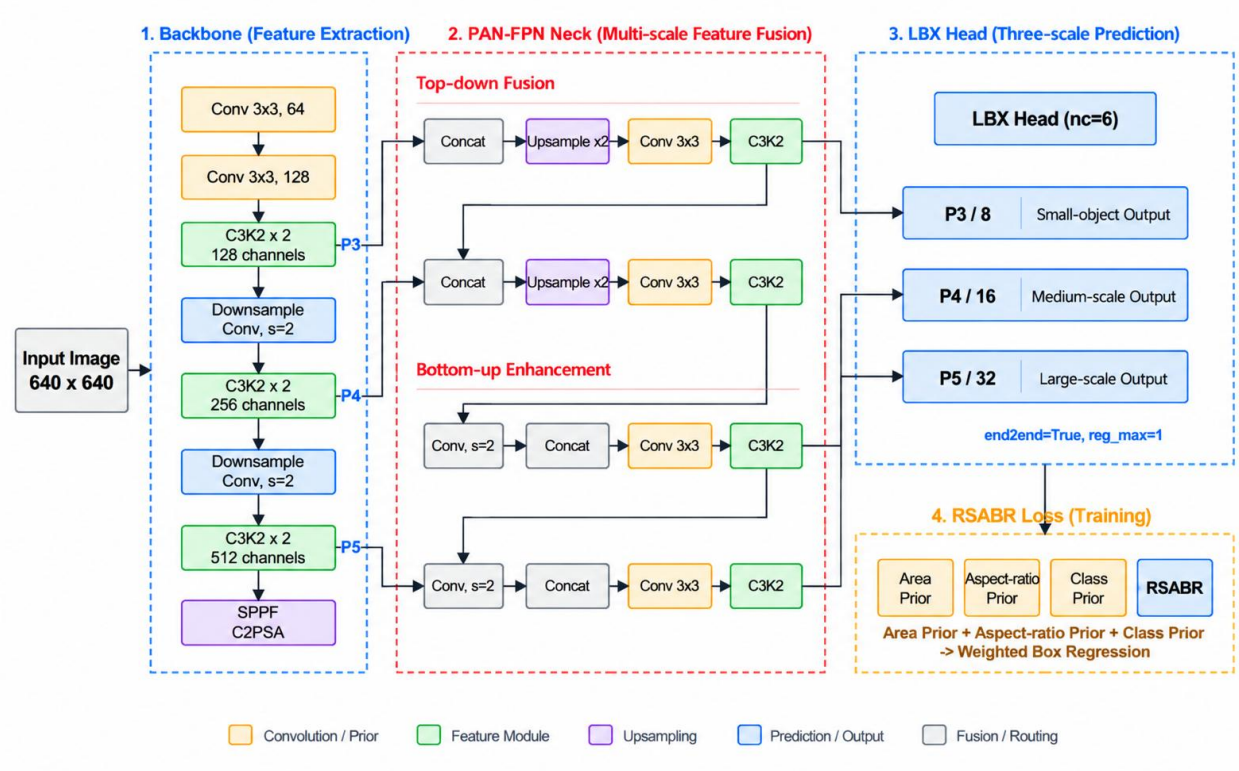

Fig. 1 Overall architecture of LBX-RSABR-YOLO26

2.2 LBX Detection Head

The LBX detection head is designed to improve the representation of rail-defect boundaries and consists of channel alignment, shared detail enhancement, classification and regression prediction, and boundary residual

correction. Its central idea is to share differential-detail features in a common latent space and to apply lightweight residual compensation to the four direct *l/t/r/b* distance outputs used by YOLO26n.

Given the input feature *P_i* at detection scale *i*, LBX first performs channel alignment with a 1 × 1 convolution, GroupNorm, and SiLU activation. Inputs from different scales are mapped to a common hidden space to obtain *H_i*, as shown in Eq. (1).

$$H_i = \phi_i(P_i) = \mathrm{SiLU}\left(\mathrm{GN}\left(\mathrm{Conv}_{1\times1}(P_i)\right)\right) \tag{1}$$

Here, *phi_i*(.) denotes the channel-alignment function at scale *i*. This operation unifies the channel dimensions of *P3*, *P4*, and *P5* and reduces distribution differences before multiscale features enter the detection head.

LBX uses a shared detail-enhancement unit *Psi*(.) to extract local edge and directional-texture features. The unit combines center-difference, horizontal-difference, vertical-difference, diagonal-difference, and standard convolutions. Its feature-enhancement process and kernel combination are given in Eqs. (2) and (3), respectively.

$$E_i = \Psi(H_i) = \sigma(K_{\mathrm{DE}} * H_i) \tag{2}$$

$$K_{\mathrm{DE}} = K_c + K_h + K_v + K_a + K_s \tag{3}$$

Here, *Kc*, *Kh*, *Kv*, *Ka*, and *Ks* denote the center-difference, horizontal-difference, vertical-difference, diagonal-difference, and standard convolution kernels, respectively. The asterisk denotes convolution, and *sigma*(.) denotes a nonlinear activation function. Through Eqs. (2) and (3), the model enhances elongated edges, local gray-level transitions, and directional textures in cracks, fractures, and other defects.

During prediction, the enhanced feature *E_i* is sent to the regression and classification branches. The main regression branch outputs the four base *l/t/r/b* distances, while the classification branch outputs class-prediction logits, as shown in Eqs. (4) and (5), respectively.

$$D_i^{\mathrm{base}} = \mathrm{Scale}_i\left(\mathrm{Conv}_{\mathrm{box}}(E_i)\right) \tag{4}$$

$$S_i = \mathrm{Conv}_{\mathrm{cls}}(E_i) \tag{5}$$

Here, *D_i*^base denotes the four base *l/t/r/b* distances at scale *i*, *S_i* denotes the classification logits, and Scale_*i*(.) is the scaling factor for the corresponding detection scale. The base regression branch primarily generates initial distances from the anchor point to the four box sides, but weak boundaries and elongated targets can still cause boundary shifts.

To further correct the bounding-box regression, LBX introduces a boundary residual-correction branch. This branch models strip context and applies channel-attention gating to extract compensatory features related to defect boundaries, and then outputs a boundary residual. The context feature and boundary residual are given in Eqs. (6) and (7), respectively.

$$Z_i = \mathrm{Strip}(E_i) \odot \mathrm{CA}(E_i) \tag{6}$$

$$R_i = \lambda \cdot \mathrm{Conv}_{1\times1}^{0}(Z_i) \tag{7}$$

Here, Strip(·) denotes horizontal and vertical strip-convolution modeling, CA(·) denotes channel-context gating, ⊙ denotes element-wise multiplication, R_i is the boundary residual correction, λ is the residual scaling coefficient, and Conv₀^(1 × 1) is a zero-initialized 1 × 1 convolution. Zero initialization keeps R_i close to zero at the start of training, allowing LBX to begin from the baseline detection head without an abrupt change in its output distribution.

Finally, LBX adds the boundary residual *R_i* to the four base distances and decodes the predicted box directly from the corrected *l/t/r/b* distances, as shown in Eqs. (8) and (9), respectively.

$$D_i = D_i^{\mathrm{base}} + R_i \tag{8}$$

$$B_i = \mathrm{dist2bbox}\left((D_i)\right) \tag{9}$$

Equations (6)-(9) show that LBX refines the four side distances before box decoding. The corrected distances are then converted directly to box coordinates, thereby alleviating local boundary shifts for elongated and weak-boundary defects while keeping the detection head lightweight. The LBX detection-head structure is shown in Fig. 2.

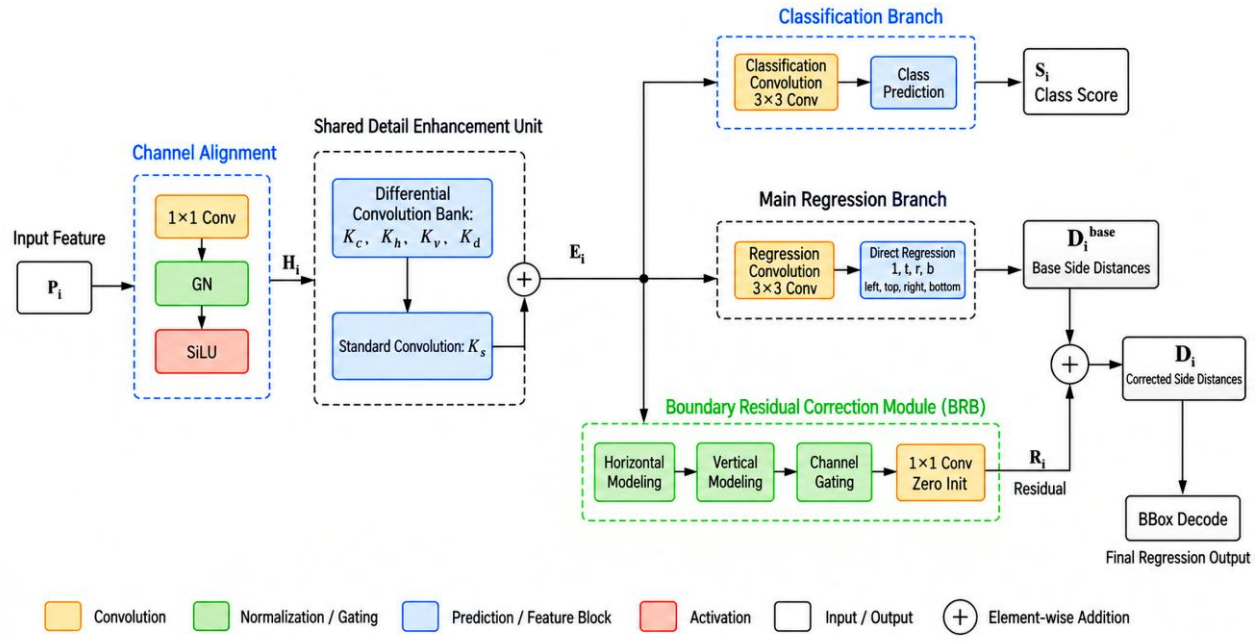

Fig. 2 LBX detection-head module

2.3 RSABR Loss

Rail defects differ substantially in area, aspect ratio, and class frequency. Cracks and fractures are usually small and elongated, so a slight pixel-level shift can cause a large IoU decrease. The scale and boundary shape of the other classes are likewise diverse, which makes a uniform regression weight a poor fit for this dataset [13].

RSABR first constructs a small-object score from the ground-truth box area. Let *w_j* and *h_j* be the width and height of the ground-truth box for positive sample *j*, and let the input image size be *W* x *H*. The area ratio *a_j* is defined in Eq. (10).

$$a_j = \frac{w_j h_j}{WH} \tag{10}$$

Based on this ratio, a small-object score *s_j* is constructed as shown in Eq. (11).

$$s_j = \mathrm{clamp}\left(\frac{\tau_s}{a_j + \tau_s}, 0{,}1\right) \tag{11}$$

Here, *tau_s* is the smoothing constant for small-object area. Equation (11) shows that as the ground-truth box becomes smaller, *a_j* decreases and *s_j* approaches 1, so the model assigns greater regression attention to small-area defects during training.

To describe defect elongation, RSABR further constructs an elongation score from the ground-truth aspect ratio. The aspect ratio *r_j* is first calculated as shown in Eq. (12).

$$r_j = \max\left(\frac{w_j}{h_j}, \frac{h_j}{w_j}\right) \tag{12}$$

The elongation score *e_j* is then obtained using the elongation threshold *tau_r*, as shown in Eq. (13).

$$e_j = \mathrm{clamp}\left(\frac{r_j}{\tau_r} - 1{,}0{,}1\right) \tag{13}$$

Here, *tau_r* is the elongation threshold. When *r_j* exceeds *tau_r*, *e_j* increases with elongation. For approximately square targets or targets with a small aspect ratio, *e_j* remains low, preventing regular-shaped targets from receiving excessive regression weight.

At the class level, RSABR defines a class-specific box weight *eta_cj* to reflect differences in localization sensitivity and sample distribution among defect classes, as shown in Eq. (14).

$$\eta_{c_j} \in [1.35, 1.55, 0.95, 1.10, 1.05, 1.00] \tag{14}$$

Here, *eta_cj* is determined by the class of positive sample *j*, with weights corresponding in order to Crack, Fracture, Corrugation, Spalling, Squat, and Wheel Burn. Crack and Fracture are generally smaller and more elongated and therefore receive greater regression attention, whereas Corrugation has a relatively stable shape and a slightly lower weight.

After combining the area, aspect-ratio, and class priors, the unnormalized shape-aware weight *omega_j* is calculated using Eq. (15).

$$\omega_j = \min\left(\gamma, \left(1 + \alpha s_j + \beta e_j\right)\eta_{c_j}\right) \tag{15}$$

To prevent large weights from destabilizing training, RSABR clips the unnormalized weights in Eq. (15) and then applies the mean normalization in Eq. (16), thereby maintaining a stable overall loss scale.

$$\hat{\omega}_j = \frac{\omega_j}{\mathrm{mean}(\omega_j)} \tag{16}$$

Finally, the normalized shape-aware weight is applied to the CIoU regression term to obtain the weighted RSABR regression loss in Eq. (17).

$$L_{\mathrm{RSABR}} = \frac{\sum_j q_j \, \hat{\omega}_j L_{\mathrm{CIoU},j}}{\sum_j q_j} \tag{17}$$

Here, *alpha* and *beta* control the gains for small and elongated targets, respectively. *Gamma* is the maximum attention weight. *q_j* is the positive-sample quality weight obtained from task assignment [14, 15]. *L_CIoU,j* is the CIoU regression loss for positive sample *j*. RSABR jointly adjusts positive-sample regression contributions using area, aspect-ratio, and class priors, directing training toward boundary localization for small, elongated, and easily confused defects. The construction of RSABR is illustrated in Fig. 3.

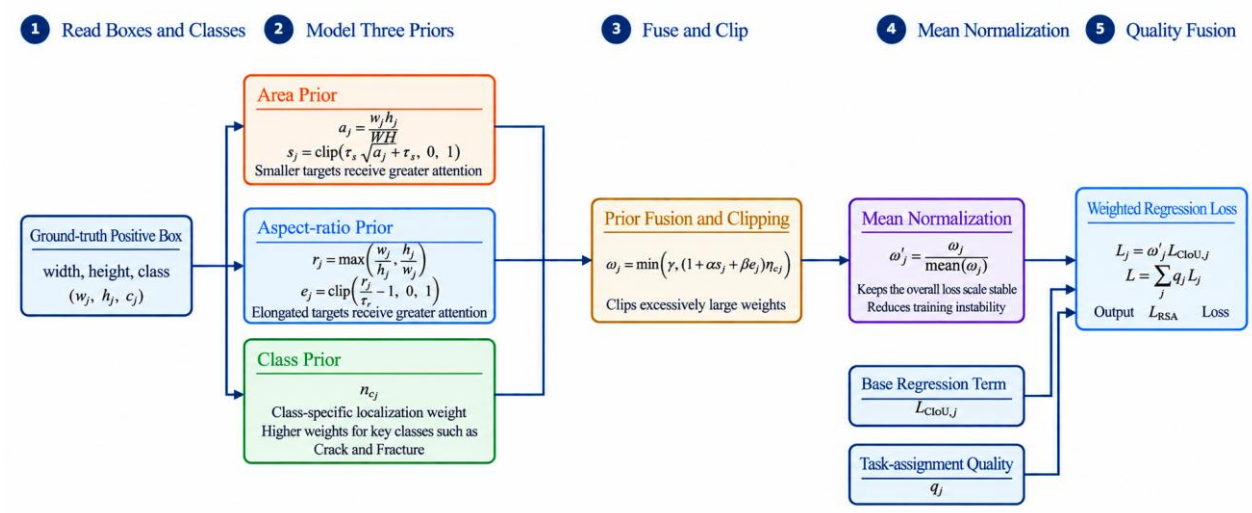

Fig. 3 Schematic construction of the RSABR loss

2.4 Overall Workflow and Module Interaction

The overall workflow of LBX-RSABR-YOLO26 is shown in Fig. 4. LBX operates in the detection head: it enhances local boundary details and applies residual correction to the four box distances before decoding. RSABR operates only during optimization, where it reweights the positive-sample regression term. The two components therefore intervene at different stages—prediction representation and training supervision—without requiring a larger backbone.

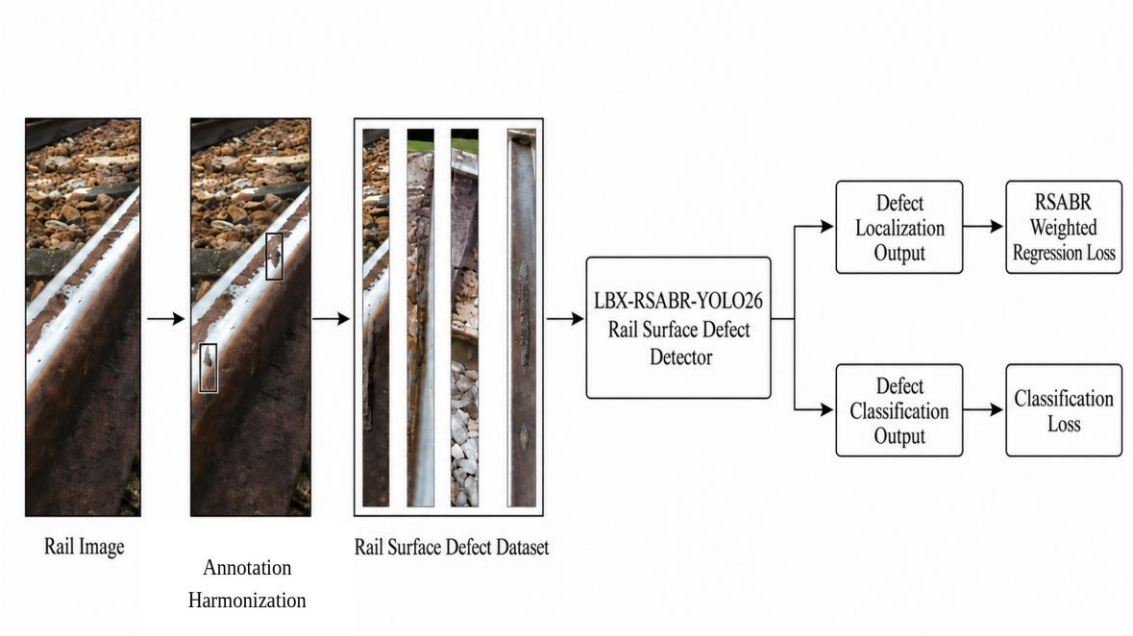

Fig. 4 Rail-surface defect detection workflow of LBX-RSABR-YOLO26

3 Experiments and Results

3.1 Dataset Sources and Data Partition

The dataset used in this study was assembled from two publicly available rail-defect datasets on Roboflow Universe: 01_FullDevelopment_RailwayDataset released by RESARCHMASTERCHULA [17] and Anotasi 1 released by YOLO REL [18], both licensed under CC BY 4.0. The source annotations were converted to a unified six-class YOLO format, and the Putus class in Anotasi 1 was mapped to Fracture. After data integration and annotation harmonization, the final prepared dataset contained 7,400 images and 12,895 annotated objects in six classes: Crack, Fracture, Corrugation, Spalling, Squat, and Wheel Burn. The final prepared dataset was divided at the image level into predefined training, validation, and test sets at a ratio of 7:2:1, containing 5,180, 1,480, and 740 images and 9,063, 2,554, and 1,278 annotated objects, respectively. The class distribution is summarized in Table 1. All comparative and ablation experiments used the same training and validation partitions.

Table 1 Class distribution of the rail-defect dataset

Class	Train	Validation	Test	Total
Crack	697	202	102	1001
Fracture	592	171	84	847
Corrugation	1375	374	189	1938
Spalling	2696	746	377	3819
Squat	2890	822	413	4125
Wheel Burn	813	239	113	1165

3.2 Experimental Setup and Evaluation Metrics

All baseline and single-module experiments use the same data split and training settings: 640 × 640 input images, 300 epochs, the SGD optimizer, an initial learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005. The YOLOv8, YOLOv11, YOLOv12, YOLO26n, and RT-DETR-S experiments are retrained under uniform experimental settings, while the LBX-only and RSABR-only experiments both use the lightweight YOLO26n configuration. Table 2 lists the experimental environment.

Table 2 Experimental environment

Configuration	Value
Operating system	Windows 11, 64-bit

Configuration	Value
GPU	NVIDIA GeForce RTX 3060 Laptop GPU
CPU	Intel Core i5-11400H CPU
CUDA	CUDA 11.2
Deep-learning framework	PyTorch

The evaluation metrics include precision P, recall R, mAP50, and mAP50-95. We additionally report parameter count, FLOPs, average forward-pass time per image, FPS, and peak GPU-memory usage to describe model complexity and throughput. Precision P measures the proportion of true defects among all predicted defects, as shown in Eq. (18).

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{18}$$

Here, TP is the number of correctly detected defect targets, and FP is the number of background or non-target instances incorrectly classified as defects. A higher P indicates fewer false detections. Recall R measures the proportion of true defects that are detected, as shown in Eq. (19).

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{19}$$

Here, FN is the number of true defect targets not detected by the model. A higher R indicates fewer missed detections. To evaluate localization quality across IoU thresholds, we use mAP50-95 as the primary metric, as shown in Eq. (20).

$$\mathrm{mAP}_{50-95} = \frac{1}{10} \sum_{t \in \{0.50, 0.55, \ldots, 0.95\}} AP_t \tag{20}$$

Here, AP_t denotes the average precision at IoU threshold t. mAP50 reflects detection performance under a relatively permissive IoU threshold, whereas mAP50-95 aggregates multiple IoU thresholds and is more sensitive to localization quality. Because elongated defects such as cracks and fractures are particularly sensitive to box shifts, mAP50-95 is used as the primary accuracy metric. Complexity and throughput metrics are reported separately to characterize the lightweight deployment trade-off under the same hardware conditions.

Table 3 summarizes model complexity, throughput-oriented inference measurements, and validation performance. Parameter count, FLOPs, average forward-pass time per image, FPS, and peak GPU memory were measured on an RTX 3060 Laptop GPU using 640 x 640 inputs and a batch size of 16. Accordingly, the reported per-image time is a batch-averaged forward-pass measurement rather than batch-1 latency. LBX-RSABR-YOLO26 contains 2.5017 M parameters and requires 6.2914 G FLOPs. Its average forward-pass time is 2.92 ms per image, corresponding to approximately 342.40 FPS, with a peak GPU memory usage of 465.17 MB. The model reaches 0.9197 mAP50, 0.7090 mAP50-95, 0.8875 precision, and 0.8758 recall at its best logged validation point.

Table 3 Model complexity, batch-averaged inference efficiency, and peak validation records

Model	Params (M)	FLOPs (G)	Time/img (ms)	FPS	VRAM (MB)	mAP50-95
YOLO26n	2.6433	6.3966	3.00	333.19	468.12	0.6219
LBX-RSABR-YOLO26	2.5017	6.2914	2.92	342.40	465.17	0.7090
RT-DETR-S	5.1893	19.1247	6.48	154.40	782.73	0.5455
LBX-RSABR-YOLO26s	9.9342	24.6649	6.55	152.79	848.42	0.7198

3.3 Baseline Comparison

To establish a lightweight baseline for subsequent improvements, we retrained RT-DETR-S [3], YOLOv8 [29], YOLOv11 [30], YOLOv12 [28], YOLO26n, and YOLO26s [16] using a common data partition and training settings. Table 4 presents the resulting validation records. Related rail-surface applications of deep-learning detectors are discussed in Refs. [6, 9-12, 19-25].

Table 4 Validation results of baseline detectors under common experimental settings

Model	*P*	*R*	mAP50	mAP50-95
RT-DETR-S	0.839	0.842	0.8509	0.5455
YOLOv11	0.875	0.835	0.8819	0.6203
YOLOv12	0.891	0.783	0.8712	0.5856
YOLOv8	0.870	0.847	0.8888	0.6169
YOLO26n	0.878	0.837	0.8881	0.6219
YOLO26s	0.874	0.872	0.9138	0.6891

Table 4 shows that YOLO26n achieves an mAP50-95 of 0.6219, slightly exceeding YOLOv11 (0.6203) and YOLOv8 (0.6169) and outperforming YOLOv12 (0.5856). YOLOv12 attains higher precision but lower recall, indicating a more conservative prediction pattern. Although YOLO26s reaches a higher mAP50-95 of 0.6891, the subsequent study focuses on YOLO26n so that the proposed modules can be assessed without increasing the baseline network size.

3.4 Main Module Ablation

Table 5 compares the lightweight YOLO26n baseline with RSABR-only, LBX-only, and combined configurations, and additionally reports a YOLO26s combination to illustrate the effect of model scale. The lightweight ablation group uses the same data split, input resolution, optimizer, and training schedule, allowing the standalone and combined effects of LBX and RSABR to be examined under consistent experimental settings.

Table 5 Peak validation records for the main LBX/RSABR ablation

Model	*P*	*R*	mAP50	mAP50-95
YOLO26n	0.878	0.837	0.8881	0.6219
YOLO26n+RSABR	0.8855	0.8635	0.9103	0.6722
YOLO26n+LBX	0.8979	0.8579	0.9126	0.6790
YOLO26n+LBX+RSABR	0.8875	0.8758	0.9197	0.7090
YOLO26s+LBX+RSABR	0.865	0.9000	0.9249	0.7198

Under the lightweight YOLO26n configuration, RSABR and LBX individually increase mAP50-95 from 0.6219 to 0.6722 and 0.6790, corresponding to gains of 5.03 and 5.71 percentage points, respectively. The combined configuration reaches 0.7090, an 8.71-point improvement over the baseline, demonstrating the complementary effect of the two mechanisms. The YOLO26s combination reaches 0.7198, but requires 9.9342 M parameters and 24.6649 G

FLOPs. These results show that the proposed lightweight configuration captures most of the accuracy benefit without the substantially larger computational cost of YOLO26s.

3.5 Class-wise Analysis and Fine-grained Ablation

Table 6 reports class-wise AP50-95 for the final combined model. The six class scores are 0.6614, 0.7454, 0.8213, 0.6311, 0.6884, and 0.7029 for Crack, Fracture, Corrugation, Spalling, Squat, and Wheel Burn, respectively, giving a six-class mean of 0.7084. All six values exceed those of YOLO26n+LBX, with Fracture showing the largest increase, from 0.6562 to 0.7454.

Table 6 Class-wise AP50-95 comparison across models

Model	Crack	Fracture	Corrugation	Spalling	Squat	Wheel Burn
YOLO26n	0.6223	0.5853	0.7487	0.5430	0.6359	0.5921
YOLO26n+RSABR	0.6484	0.6477	0.7988	0.5989	0.6792	0.6616
YOLO26n+LBX	0.6549	0.6562	0.8016	0.6107	0.6842	0.6694
YOLO26n+LBX+RSABR	0.6614	0.7454	0.8213	0.6311	0.6884	0.7029

Table 7 isolates the internal LBX components within the combined framework. Relative to the complete configuration (0.7090 mAP50-95), removing strip context, boundary residual correction, or zero initialization reduces mAP50-95 to 0.6725, 0.6713, and 0.6670, respectively. The largest drop occurs when zero initialization is removed, while suppressing the boundary residual branch also causes a substantial decrease. These results support the joint use of strip context, boundary-distance residual correction, and zero-initialized residual learning in LBX.

Table 7 Peak validation ablation of LBX internal components

Model	Strip context	Boundary residual correction	Zero initialization	mAP50	mAP50-95
YOLO26n	×	×	×	0.8881	0.6219
w/o strip context	×	√	√	0.9122	0.6725
w/o boundary residual correction	√	×	√	0.9092	0.6713
w/o zero initialization	√	√	×	0.9073	0.6670
LBX-RSABR	√	√	√	0.9197	0.7090

Table 8 shows that applying the area prior alone increases mAP50–95 from 0.6219 to 0.6692. Adding the aspect-ratio prior and the class prior changes the values to 0.6688 and 0.6687, respectively, so these priors do not provide additional gains when introduced independently. Their effectiveness becomes evident when they are integrated with clipping, mean normalization, and task-assignment quality weighting in the complete RSABR formulation. The notation "+A+AR+C / (w/o Norm & q)" indicates that all three priors are used without clipping, mean normalization, or positive-sample quality weighting. The complete RSABR model achieves an mAP50–95 of 0.6722, showing the benefit of combining the priors with the full weighting procedure.

Table 8 Peak validation ablation of RSABR weighting components

Model	Area prior	Aspect-ratio prior	Class prior	mAP50	mAP50-95
YOLO26n	×	×	×	0.8881	0.6219
+Area	√	×	×	0.9080	0.6692

Model	Area prior	Aspect-ratio prior	Class prior	mAP50	mAP50-95
+Area+AR	√	√	×	0.9067	0.6688
+A+AR+C (w/o Norm & q)	√	√	√	0.9069	0.6687
RSABR	√	√	√	0.9103	0.6722

Taken together, Tables 6-8 provide complementary evidence at the class and component levels. The final combined model achieves higher AP50-95 for all six classes than either single-module configuration, while the internal ablations show that removing any listed LBX component or simplifying the complete RSABR weighting mechanism reduces performance.

3.6 Qualitative Results and Training Dynamics

Figure 5 provides qualitative visualizations of representative outputs from four models on the same scene. For the displayed Spalling and Squat examples, the final combined model assigns higher confidence scores than the comparison models while producing similar box locations. Quantitative comparisons are reported in Tables 4–6.



Fig. 5 Example comparison of detection results

Figures 6-8 show the mAP50, precision, and recall trajectories of the four lightweight configurations over 300 epochs. The curves overlap and intersect frequently during early training, but the combined configuration more consistently occupies the upper range during the middle and late stages. The baseline generally remains lower, whereas the two single-module configurations show intermediate behavior.

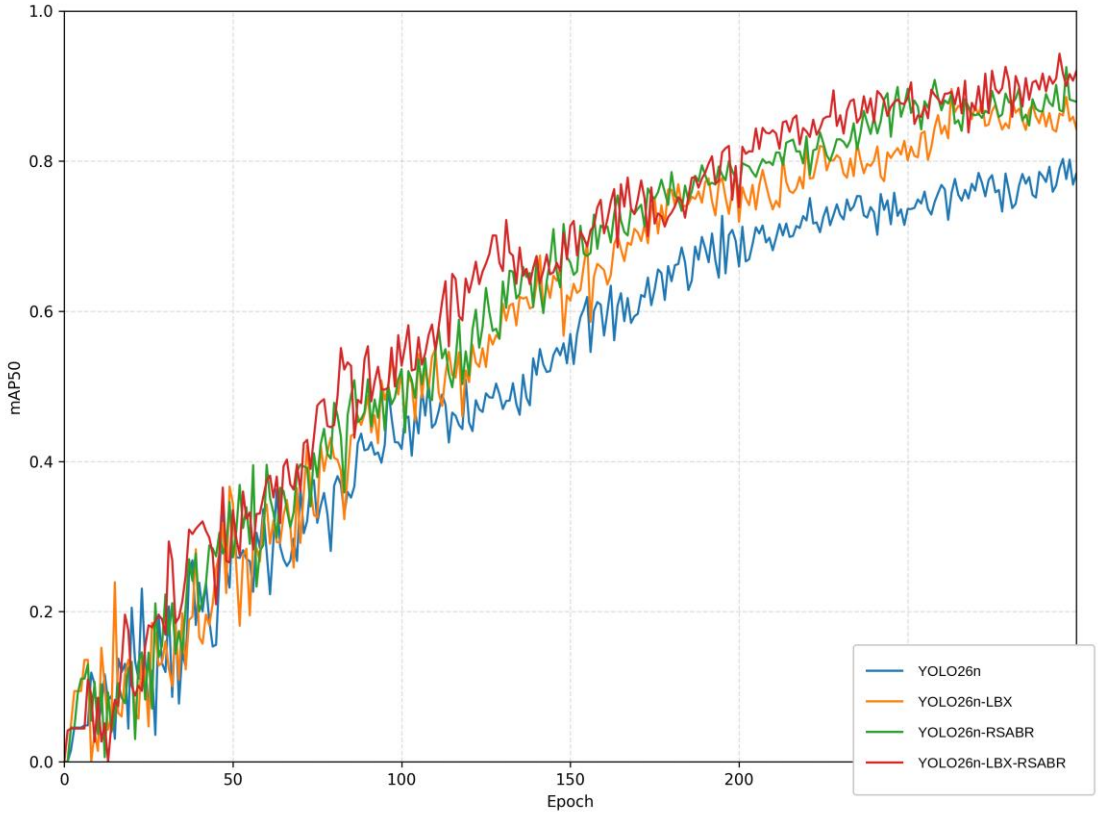

Fig. 6 Comparison of mAP50 training curves for the lightweight ablation group

Figures 7 and 8 further show that precision and recall fluctuate across epochs for all configurations. During late training, the combined model generally remains in the upper range for both metrics. These curves illustrate the optimization behavior of the four configurations over the training schedule.

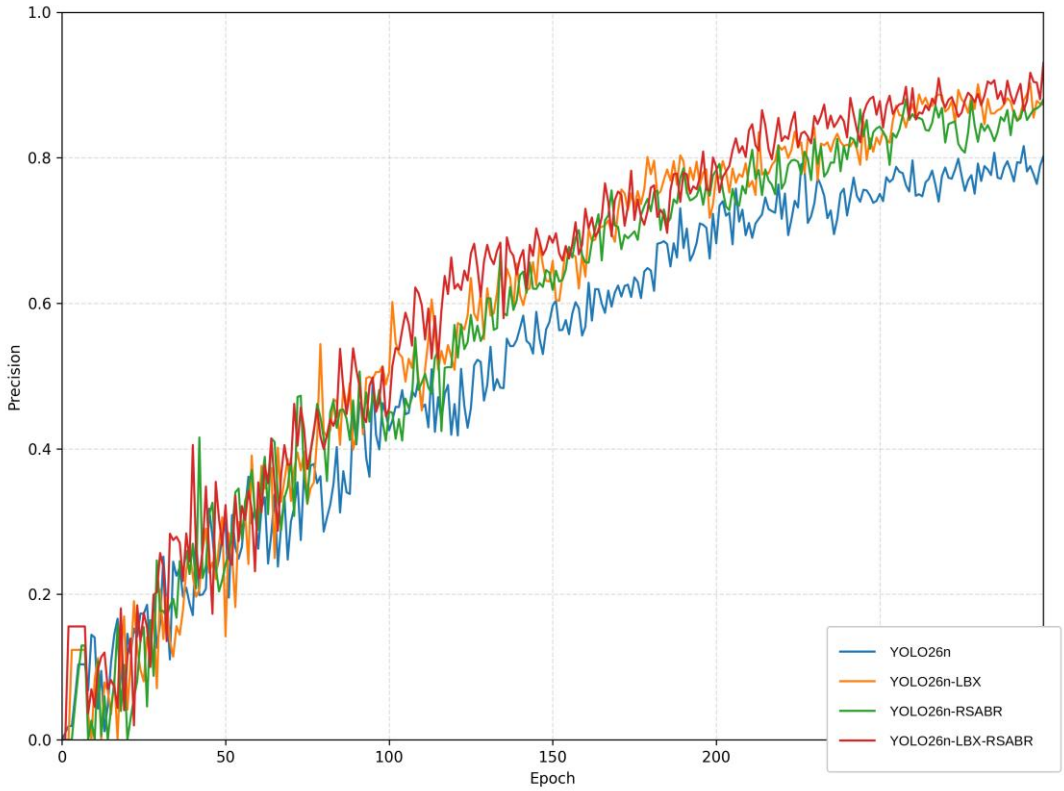

Fig. 7 Comparison of precision training curves for the lightweight ablation group

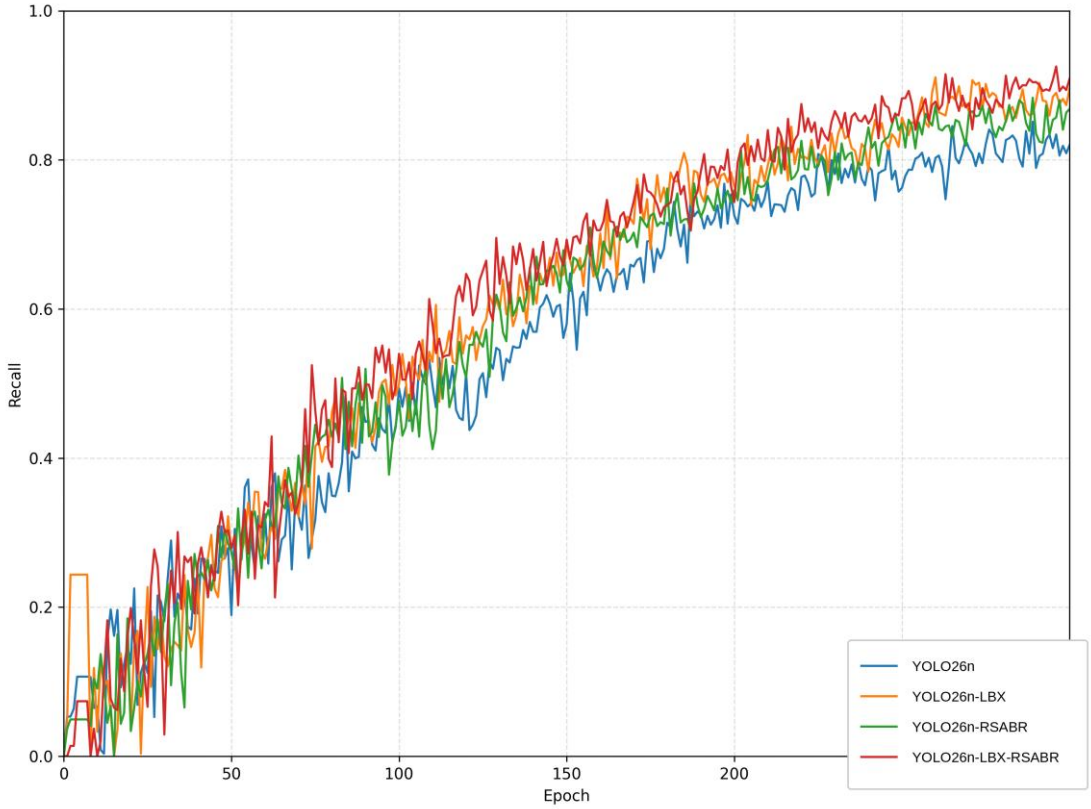

Fig. 8 Comparison of recall training curves for the lightweight ablation group

4 Discussion

The experimental results support two main observations. First, LBX is the stronger standalone contribution in the lightweight YOLO26n setting: mAP50-95 increases by 5.71 percentage points with LBX and by 5.03 points with RSABR. When the two mechanisms are combined, the largest class-wise improvement occurs for Fracture, which rises from 0.6562 with YOLO26n+LBX to 0.7454. This pattern directly supports the design goal of improving localization for weak and elongated boundaries while preserving a compact detector.

Second, the accuracy improvement is achieved with a smaller lightweight model. Compared with YOLO26n, the final model reduces the parameter count from 2.6433 M to 2.5017 M and FLOPs from 6.3966 G to 6.2914 G, while the batch-averaged forward-pass measurement changes from 3.00 to 2.92 ms per image and throughput from 333.19 to 342.40 FPS. These measurements indicate a favorable accuracy-complexity trade-off on the reported GPU platform. The RSABR ablation shows that the area prior provides the clearest standalone improvement, whereas adding the aspect-ratio and class priors alone does not increase mAP50-95. The complete RSABR configuration achieves the best result when these priors are combined with clipping, mean normalization, and task-assignment quality weighting. The consistent training and evaluation settings make the accuracy, efficiency, and ablation results directly comparable.

5 Conclusions

This study presents LBX-RSABR-YOLO26, a lightweight rail surface defect detector that combines boundary-distance refinement with shape-aware regression reweighting. The model achieves 70.84% mAP50-95, with a peak validation record of 70.90%. Relative to YOLO26n, it reduces the parameter count from 2.6433 M to 2.5017 M and FLOPs from 6.3966 G to 6.2914 G. LBX provides the larger standalone gain, while the combined configuration delivers the strongest lightweight performance. Fine-grained analysis further shows that the boundary residual design, zero initialization, and complete RSABR weighting mechanism each contribute to the final result. Overall, the findings demonstrate the value of combining prediction-side boundary correction with training-side regression reweighting for efficient rail surface defect detection.

Statements and Declarations

Funding

No funding was received for conducting this study.

Author Contributions

YANG Chuang: conceptualization, methodology, software, investigation, data curation, visualization, and writing - original draft. LI Hua: supervision, validation, methodology, and writing - review and editing. Both authors read and approved the final manuscript.

Competing Interests

The authors declare that they have no competing interests.

Data Availability Statement

The original datasets used in this study are publicly available on Roboflow Universe, including 01_FullDevelopment_RailwayDataset [17] and Anotasi 1 [18], both released under the CC BY 4.0 license. The datasets were further harmonized in terms of annotation format and class labels for the experiments conducted in this study. Detailed information on the data sources and dataset partitioning is provided in Section 3.1.

References

[1] Wang, C.Y., Yeh, I.H., Liao, H.Y.M.: YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In: Computer Vision - ECCV 2024, pp. 1-21. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-72751-1_1

[2] Wang, A., Chen, H., Liu, L., et al.: YOLOv10: Real-Time End-to-End Object Detection. In: Advances in Neural Information Processing Systems 37, pp. 107984-108011 (2024). https://doi.org/10.52202/079017-3429

[3] Zhao, Y., Lv, W., Xu, S., et al.: DETRs Beat YOLOs on Real-Time Object Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16965-16974 (2024). https://doi.org/10.1109/CVPR52733.2024.01605

[4] Wu, Y., Qin, Y., Qian, Y., et al.: Hybrid Deep Learning Architecture for Rail Surface Segmentation and Surface Defect Detection. Computer-Aided Civil and Infrastructure Engineering 37(2), 227-244 (2022). https://doi.org/10.1111/mice.12710

[5] Wang, M., Li, K., Zhu, X., et al.: Detection of Surface Defects on Railway Tracks Based on Deep Learning. IEEE Access 10, 126451-126465 (2022). https://doi.org/10.1109/ACCESS.2022.3224594

[6] Wang, H., Li, M., Wan, Z.: Rail Surface Defect Detection Based on Improved Mask R-CNN. Computers & Electrical Engineering 102, 108269 (2022). https://doi.org/10.1016/j.compeleceng.2022.108269

[7] Zhang, C., Xu, D., Zhang, L., et al.: Rail Surface Defect Detection Based on Image Enhancement and Improved YOLOX. Electronics 12(12), 2672 (2023). https://doi.org/10.3390/electronics12122672

[8] Ge, Z., Liu, S., Wang, F., et al.: YOLOX: Exceeding YOLO Series in 2021. arXiv (2021). https://doi.org/10.48550/arXiv.2107.08430

[9] Wang, Y., Zhang, K., Wang, L., et al.: An Improved YOLOv8 Algorithm for Rail Surface Defect Detection. IEEE Access 12, 44984-44997 (2024). https://doi.org/10.1109/ACCESS.2024.3380009

[10] Cao, Y., Ma, L., Sun, Y., et al.: Improved YOLOv8 for High-Precision Detection of Rail Surface Defects on Heavy-Haul Railways. Chinese Journal of Electronics 34(3), 802-815 (2025). https://doi.org/10.23919/CJE.2024.00.200

[11] Du, J., Zhang, R., Gao, R., et al.: RSDNet: A New Multiscale Rail Surface Defect Detection Model. Sensors 24(11), 3579 (2024). https://doi.org/10.3390/s24113579

[12] Guo, F., Qian, Y., Yu, H.: Automatic Rail Surface Defect Inspection Using the Pixelwise Semantic Segmentation Model. IEEE Sensors Journal 23(13), 15010-15018 (2023). https://doi.org/10.1109/JSEN.2023.3280117

[13] Zheng, Z., Wang, P., Liu, W., et al.: Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In: Proceedings of the AAAI Conference on Artificial Intelligence 34(7), pp. 12993-13000 (2020). https://doi.org/10.1609/AAAI.V34I07.6999

[14] Li, X., Wang, W., Wu, L., et al.: Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. In: Advances in Neural Information Processing Systems 33, pp. 21002-21012 (2020). https://doi.org/10.48550/arXiv.2006.04388

[15] Feng, C., Zhong, Y., Gao, Y., et al.: TOOD: Task-Aligned One-Stage Object Detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3490-3499. IEEE, Montreal (2021). https://doi.org/10.1109/ICCV48922.2021.00349

[16] Jocher, G., Qiu, J., Liu, M., et al.: Ultralytics YOLO26: Unified Real-Time End-to-End Vision Models. arXiv (2026). https://doi.org/10.48550/arXiv.2606.03748

[17] RESARCHMASTERCHULA: 01_FullDevelopment_RailwayDataset. Roboflow Universe (2022). https://universe.roboflow.com/resarchmasterchula/01_fulldevelopment_railwaydataset. Accessed 18 September 2026. License: CC BY 4.0.

[18] YOLO REL: Anotasi 1. Roboflow Universe (2023). https://universe.roboflow.com/yolo-rel/anotasi-1. Accessed 18 September 2026. License: CC BY 4.0.

[19] Guo, F., Liu, J., Qian, Y., et al.: Rail Surface Defect Detection Using a Transformer-Based Network. Journal of Industrial Information Integration 38, 100584 (2024). https://doi.org/10.1016/j.jii.2024.100584

[20] Zhong, Y., Chen, G.: Rail Surface Defect Detection Based on Dual-Path Feature Fusion. Electronics 13(13), 2564 (2024). https://doi.org/10.3390/electronics13132564

[21] Zhang, Y., Feng, T., Song, Y., et al.: An Improved Target Network Model for Rail Surface Defect Detection. Applied Sciences 14(15), 6467 (2024). https://doi.org/10.3390/app14156467

[22] Si, C., Luo, H., Han, Y., et al.: Rail-STrans: A Rail Surface Defect Segmentation Method Based on Improved Swin Transformer. Applied Sciences 14(9), 3629 (2024). https://doi.org/10.3390/app14093629

[23] Xu, C.Y., Liao, Y., Liu, Y., et al.: Lightweight Rail Surface Defect Detection Algorithm Based on an Improved YOLOv8. Measurement 242, 115922 (2025). https://doi.org/10.1016/j.measurement.2024.115922
[24] Zhu, W., Han, X., Zhang, K., et al.: Application of YOLO11 Model with Spatial Pyramid Dilation Convolution (SPD-Conv) and Effective Squeeze-Excitation (EffectiveSE) Fusion in Rail Track Defect Detection. Sensors 25(8), 2371 (2025). https://doi.org/10.3390/s25082371
[25] Liu, Y., Wei, B., Zhang, Q., et al.: A Study on Surface Defect Detection of Metro Tunnel Contact Rails Using an Improved YOLOv11 Framework. Measurement 281, 121940 (2026). https://doi.org/10.1016/j.measurement.2026.121940
[26] Choi, J.Y., Han, J.M.: Deep Learning (Fast R-CNN)-Based Evaluation of Rail Surface Defects. Applied Sciences 14(5), 1874 (2024). https://doi.org/10.3390/app14051874
[27] Bai, T., Gao, J., Yang, J., et al.: A Study on Railway Surface Defects Detection Based on Machine Vision. Entropy 23(11), 1437 (2021). https://doi.org/10.3390/e23111437
[28] Tian, Y., Ye, Q., Doermann, D.: YOLOv12: Attention-Centric Real-Time Object Detectors. In: Advances in Neural Information Processing Systems 38, pp. 87151-87175 (2025). https://doi.org/10.52202/085713-2627
[29] Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO. Version 8.0.0 (2023). https://github.com/ultralytics/ultralytics. Accessed 18 September 2026.
[30] Jocher, G., Qiu, J.: Ultralytics YOLO11. Version 11.0.0 (2024). https://github.com/ultralytics/ultralytics. Accessed 18 September 2026.